

### 3.3 SEQUÊNCIA SIMPLES

#### PORTUGOL

Comando 1;  
Comando 2;

Comando n;

#### PASCAL

Comando 1;  
Comando 2;

Comando n;

A junção do ; (ponto e vírgula) no PORTUGOL é o equivalente ao fim de linha no PASCAL. O ; indica a separação de comandos.

### 4) ALTERNATIVA (simples e composta)

#### PORTUGOL

```
se <condição>
então <lista de comandos>
fim se
```

```
se <condição>
então <lista de comandos1>
senão <lista de comandos2>
fim se
```

#### PASCAL

```
if <expressão1>
then <comando>
if <expressão>
then <comando1>
else <comando2>
```

Observe que depois da palavra *then* ou da palavra *else* só podemos ter um comando. Se houver mais de um comando, será necessário utilizar um comando composto, isto é, colocar os comandos entre as palavras *begin* e *end*.

Como exemplo, para passar para o PASCAL.

```
se A < B
então A ← A + 2;
senão B ← B - 1;
N ← M;
fim se
```

### EXERCÍCIO

```
3. Se  $x = 10$ 
então  $A = A + 7$ 
senão  $\begin{cases} B = B - 1 \\ N = M \end{cases}$ 
fim se
```

Observe que o trecho do *if* não pode aparecer um ponto e vírgula (;) à esquerda do comando. Não é necessário o *do* (do inglês), embora, por quem gosta de estilo, seja comum na prática dos programadores PASCAL.

Para passar para o PASCAL:

```
if A < B
then A = A + 2;
B = 3;
fim se
```

### EXERCÍCIO

```
if A < B
then begin
A ← A + 2;
B ← 3;
end
```

Observe que *if* é mapeado por *<*.

Considerando o trecho de programa PASCAL a seguir, que valores terão B e C depois de executados os comandos?

```
B ← 0;
C ← 0;
P ← 3;
if P < 3 then begin
B ← 5;
C ← 10;
end
B ← B + 1;
C ← C + 1;
```

Solução:

B terá o valor 6 e C terá o valor 11. Observe que a indentação no PASCAL não é necessária pelo compilador e o trecho acima equivale a: